

libre pour table des matiere

Presentation

La minilib C est une bibliotheque C, basée sur le code de la newlib de redhat, mais fortement reduite en fonctionnalite et taillee dans la masse pour etre la plus petite possible. En ce sens, elle n'embarque pas toutes les fonctions proposees par la bibliotheque C classique (libc): voir la chapitre xxx pour connaître la liste des fonctions incluses.

La minilib necessite, pour certaines fonctions (tel printf) un « support systeme », en gros, un moyen, dependant de la plateforme et de l'application, pour realiser sa tache. Dans le cas du printf, c'est un endroit où ecrire le texte (une liaison serie le plus souvent). Pour etre le plus souple et adaptable possible, ce support systeme est volontairement non code et aboutit dans des fonctions contenue dans le fichier syscalls.c, non inclus dans la bibliotheque et que l'utilisateur doit integrer dans son projet, puis l'adapter à son besoin. Pour plus d'info, se referer au chapitre consacré aux syscalls.

Limite de ce document

Ce document se limite à la version 1.00 de la minilib.

Fonctions integrees à la minilib.

Les fonctionnalites suivent le decoupage existant au sein de la libc et que l'on retrouve dans les fichiers en-tete tel que stdio.h ou stdlib.h.

Pour plus d'information concernant les fonctions listees ici, se referer à la norme ISO/IEC C'99

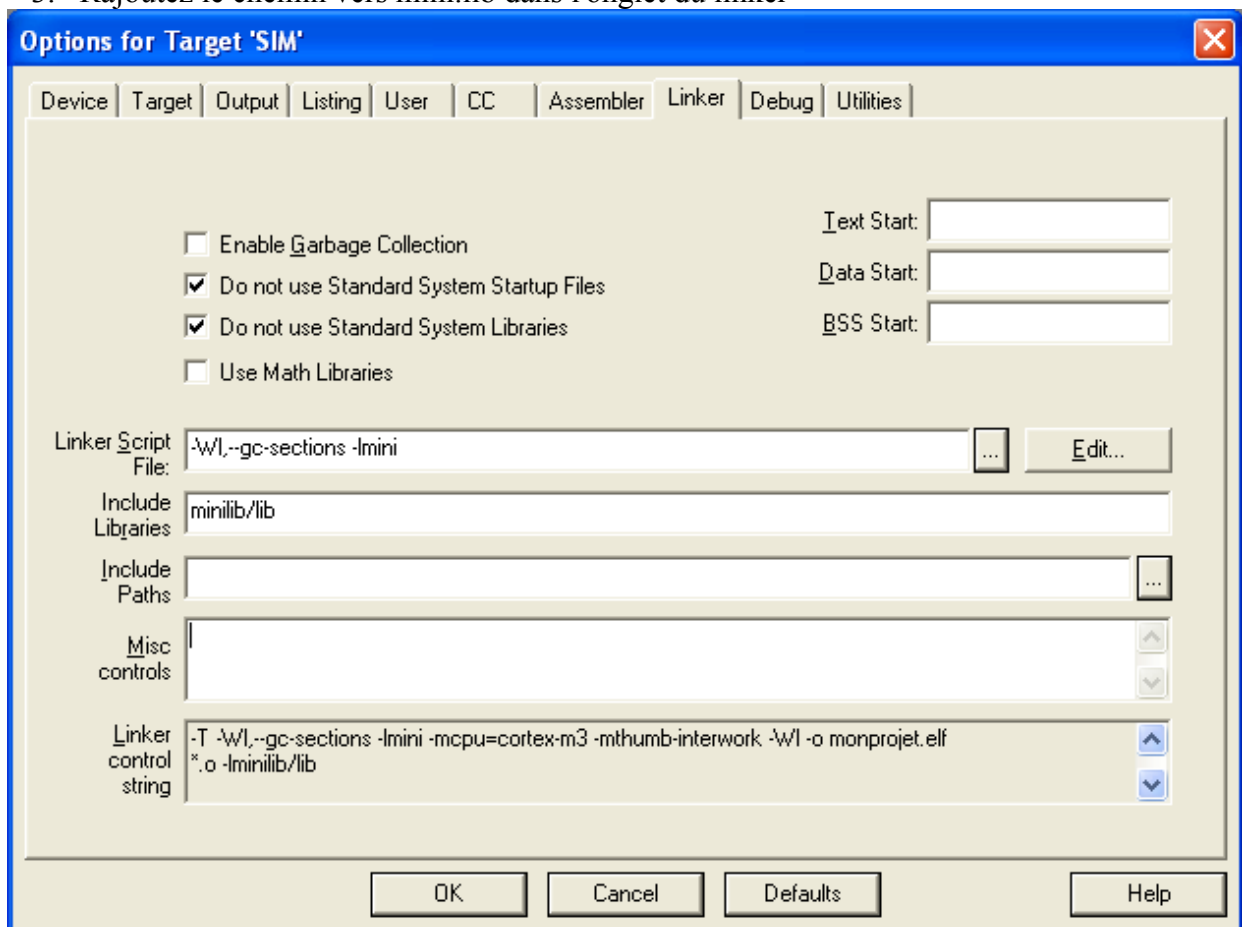
- | | | |
|---|--|--|
| <ul style="list-style-type: none">• STDIO<ul style="list-style-type: none">• printf• fprintf• sprintf• sscanf• fputc• fgetc• fputs• fgets• getchar• putchar | <ul style="list-style-type: none">• STDLIB<ul style="list-style-type: none">• atoi• atol• strtol• div• ldiv• abs• exit• assert• rand• malloc• calloc• free | <ul style="list-style-type: none">• CTYPE<ul style="list-style-type: none">• isalnum• isalpha• isascii• isblank• iscntrl• isdigit• islower• isprint• ispunct• isspace• isupper• isxdigit• toascii• tolower• toupper |
| <ul style="list-style-type: none">• STRING<ul style="list-style-type: none">• bcmp• bcopy• bzero• index• memccpy• memchr• memcmp• memcpy• memmove | <ul style="list-style-type: none">• STRING (suite)<ul style="list-style-type: none">• memcpy• memsetrindex• strcat• strchr• strcmp• strcoll• strcpy• strcspn• strlcat | <ul style="list-style-type: none">• STRING (fin)<ul style="list-style-type: none">• strlcpy• strlen• strlwr• strncat• strncmp• strncpy• strnlen• strrchr• strsep |

- strspn
- strstr
- strupr

Utilisation de la bibliotheque.

- Sous KEIL
 - Utilisation avec GCC

1. Rajoutez le fichier syscalls.c à votre projet (le template se trouve sous minilib/syscalls_template/syscalls.c)
2. Adaptez les fonctions du fichier a votre besoin (voir le chapitre xxx a propos des fonctions)
3. Rajoutez le chemin vers mini.lib dans l'onglet du linker

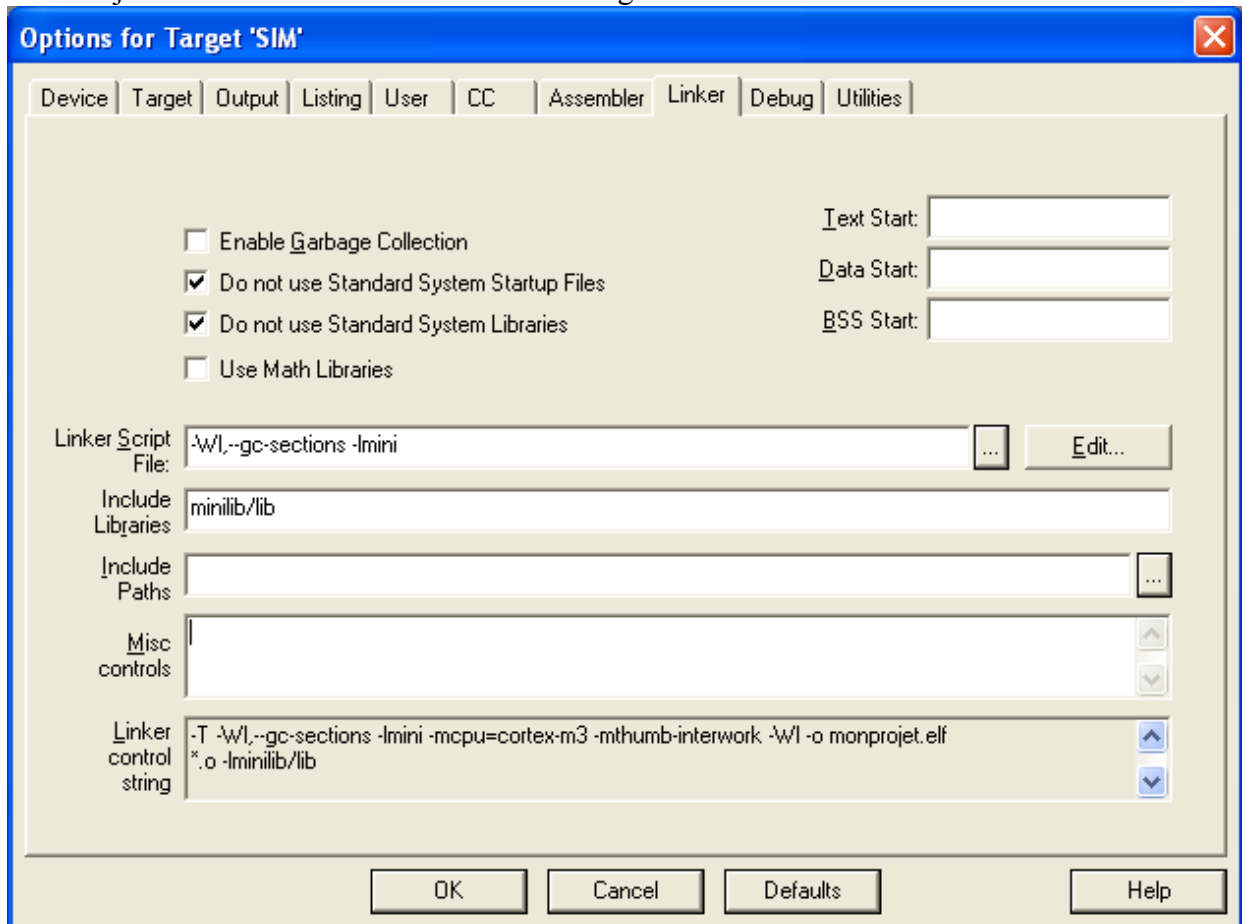


4. Bien verifier dans la fenetre precedente que les bibliotheques standards et les fichiers startup standards ne sont pas utilisés.

- Utilisation avec ARMCC

5. Rajoutez le fichier syscalls.c à votre projet (le template se trouve sous minilib/syscalls_template/syscalls.c)
6. Adaptez les fonctions du fichier a votre besoin (voir le chapitre xxx a propos des fonctions)

7. Rajoutez le chemin vers mini.lib dans l'onglet du linker



8. Bien vérifier dans la fenêtre précédente que les bibliothèques standards et les fichiers startup standards ne sont pas utilisés.

Recompiler la bibliothèque

Normalement, il n'est pas nécessaire de recompiler la bibliothèque. Néanmoins, si c'est nécessaire, le répertoire minilib/keil contient deux projets: un pour ARMCC, l'autre pour GCC. Il suffit d'ouvrir le projet ad-hoc et lancer la recompilation du projet (vérifier que la cible est bien la bibliothèque)

Sous GCC, il est très important de vérifier que les options `-ffunctions-section` et `-fdata-sections` sont bien activées: sans cela, au moment de l'édition de lien, GCC ne pourra pas supprimer les fonctions non utilisées → code gros, sans nécessité.

Options utilisables lors de la compilation de la bibliothèque

Les clés de compilation suivantes (define) sont utilisables avec la bibliothèque:

- `PREFER_SIZE_OVER_SPEED`: Utilisée essentiellement par les fonctions du répertoire string, positionnée, la clé permet d'obtenir un code très compact, mais plus lent.

Description des fonctions contenues dans syscalls.c

- `_exit`: Utilisé essentiellement par la fonction `exit()`. Par défaut contient une boucle infinie.
- `_kill`: Non utilisé pour l'instant
- `_open`: Non utilisé pour l'instant
- `_read`: Utilisé par toutes les fonctions lisant dans un flux (`fgetc`, `fgets`, ...). Renvoie un tableau

- d'octet pointé par *ptr, d'une longueur len, remplit avec les données du flux file.
- `_write`: Utilisé par toutes les fonctions écrivant dans un flux (`fputc`, `printf`, `fprintf`, ...).
Transfert un tableau pointé par *ptr, d'une longueur len, dans le flux file.
- `_malloc`: Sert à implémenter le mécanisme d'allocation mémoire
- `_free`: Compagnon de `_malloc`

Un mot au sujet des fichiers utilisant le type FILE